(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 March 2001 (15.03.2001)

PCT

(10) International Publication Number
WO 01/19074 A1

(51) International Patent Classification⁷: H04N 5/00

(21) International Application Number: PCT/US00/24097

(22) International Filing Date:
1 September 2000 (01.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/152,385 3 September 1999 (03.09.1999) US

(71) Applicant (*for all designated States except US*): GENERAL INSTRUMENT CORPORATION [US/US]; 101 Tournament Drive, Horsham, PA 19044 (US).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): SPRUNK, Eric, J. [US/US]; 6421 Cayenne Lane, Carlsbad, CA 92006 (US).

(74) Agents: FRANKLIN, Thomas, D. et al.; Townsend and Townsend and Crew LLP, Two Embarcadero Center, Eighth Floor, San Francisco, CA 94111-3834 (US).

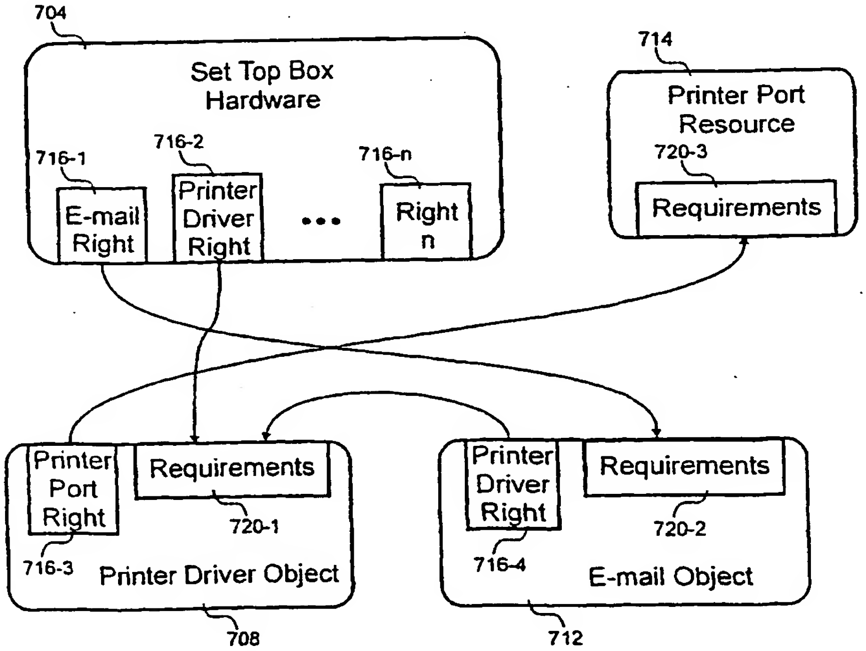
(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

[Continued on next page]

(54) Title: ENTITLEMENTS OF OBJECTS AND RESOURCES



(57) **Abstract:** The invention relates to securing information in a secure access system. In one embodiment a method secures information in a conditional access system to authorize a functional unit. The functional unit has requirements related to the functional unit. A transmission conduit is entitled to a content receiver. Rights related to the functional unit are received. The rights are correlated with the requirements. The rights are checked against the requirements.

WO 01/19074 A1

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

ENTITLEMENTS OF OBJECTS AND RESOURCES

This application claims the benefit of U.S. Provisional Application No. 60/152,385 filed on September 3, 1999.

BACKGROUND OF THE INVENTION

This invention relates in general to secure access systems and, more specifically, to securing information within the secure access systems.

Cable television (TV) providers distribute video streams to subscribers by way of conditional access (CA) systems. CA systems distribute video streams from a headend of the cable TV provider to a set top box associated with a subscriber. The headend includes hardware that receives the video streams and distributes them to the set top boxes within the CA system. Select set top boxes are allowed to decode certain video streams according to entitlement information sent by the cable TV provider to the set top box.

Video programs are distributed in either digital form or analog form to the set top boxes. There are around one hundred and twenty analog carrier channels in most cable television systems. The carrier channels either carry an analog video stream or carry multiple digital video streams. The analog video feed is modulated on a carrier and occupies the whole carrier channel for the one analog video feed. To maximize bandwidth, about eight to fourteen digital video streams may be statistically multiplexed on the same carrier channel. The separate digital video streams are separated by packet identification (PID) information such that the individual content streams can be removed according to their unique PID information.

Video streams are broadcast to all set top boxes, but only a subset of those boxes is given access to specific video streams. For example, only those that have ordered a pay per view boxing match are allowed to view it even though every set top box may receive the match in encrypted form. Once a user orders the pay per view program, an entitlement message is broadcast in encrypted form to all set top boxes. Only the particular set top box the entitlement message is intended for can decrypt it. Inside the decrypted entitlement message is a key that will decrypt the pay per view program. With that key, the set top box decrypts the pay per view program as it is received in real-time as either an analog or digital video stream. Accordingly, only whole video streams are entitled during download.

Some systems, that do not provide conditional access, integrate personal computing with a TV for displaying programs. For example, products such as WebTV™ integrate web browsing and e-mail programs with a TV. In these systems, a personal computer (PC) is housed near the TV. The PC is connected to an Internet service provider (ISP) that provides the content for the web browsing and e-mail programs. These systems provide content without checking entitlements as is required in conditional access systems.

SUMMARY OF THE INVENTION

The invention relates to securing information in a secure access system. In one embodiment a method secures information in a conditional access system to authorize a functional unit. The functional unit has requirements related to the functional unit. A transmission conduit is entitled to a content receiver. Rights related to the functional unit are received. The rights are correlated with the requirements. The rights are checked against the requirements to provide authorization of the functional unit.

Reference to the remaining portions of the specification, including the drawings and claims, will realize other features and advantages of the present invention. Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with respect to the accompanying drawings. In the drawings, like reference numbers indicate identical or functionally similar elements.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing one embodiment of a content delivery system;

Fig. 2 is a block diagram illustrating an embodiment of a set top box interfaced to its environment;

Fig. 3 is a block diagram depicting an embodiment of an object message;

Fig. 4 is a block diagram showing an embodiment of a "rights" message;

Fig. 5 is a block diagram showing an embodiment of a "requirements" message;

Fig. 6 is a block diagram showing the relationship between different objects in a set top box;

Fig. 7 is a block diagram illustrating an embodiment of interaction between functional units;

Fig. 8 is a flow diagram showing an embodiment of a process for distributing functional units;

Fig. 9 is a flow diagram depicting an embodiment of a process for receiving functional units and authorizing them;

Fig. 10 is a flow diagram illustrating an embodiment of a process for authenticating and authorizing a software object;

Fig. 11 is a flow diagram illustrating an embodiment of a process for authenticating and authorizing a running software object; and

Fig. 12 is a flow diagram illustrating another embodiment of a process for authenticating and authorizing a running software object.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

The present invention validates that functional units, such as software, are authorized for use within a television (TV) set top box. Objects and resources are two examples of functional units. After the functional unit is installed in the set top box, authorization and/or authentication checks are performed when checkpoints are encountered. Checkpoints are triggered, for example, when the functional units in the set top box interact with each other.

In the Figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

Referring first to Fig. 1, a block diagram of one embodiment of a content delivery system 100 is shown. The delivery system 100 selectively provides content to a number of users based upon certain conditions being satisfied. Included in the system 100 are a headend 104, number of set top boxes 108, local programming receiver 112, satellite dish 116, and the Internet 120.

The headend 104 receives content and distributes that content to users. Content can include video, audio, interactive video, software, firmware, and/or data. This content is received from a variety of sources that could include the satellite dish 116, the local programming receiver 112, a microwave receiver, a packet switched network, the Internet 120, etc. Each set top box 108 has a unique address that allows sending entitlement information to an individual set top box 108. In this way, one set top box 108-1 might be entitled to some particular content while another 108-2 might not. Equipment within the headend 104 regulates the subset of set top boxes 108 are entitled to some particular content.

The content is generally distributed in digital form through an analog carrier channel that contains multiple content streams. All the content streams are multiplexed together into a digital stream that is modulated upon the analog carrier channel. The separate content streams are tracked by packet identification (PID) information such that the individual content streams can be removed according to their unique PID information. There are around one hundred and twenty analog carrier channels in this embodiment of the system 100. Other embodiments could distribute the content with transport mechanisms that include satellite dishes, microwave antennas, RF transmitters, packet switched networks, cellular data modems, carrier current, phone lines, and/or the Internet.

Referring next to Fig. 2, a block diagram of an embodiment of a display system 200 is shown. This embodiment provides multiple levels of object and resource security through a variety of security mechanisms. Included in the display system 200 are a set top box 108, network 208, printer 212, TV display 216, and wireless input device 218. These items cooperate in such a way that the user can enjoy content conditionally distributed by a content provider. The content can include video, audio, software, firmware, interactive TV, data, text, and/or other information. In this embodiment, the content provider is a cable TV provider or multiple system operator (MSO).

The network 208 serves as the conduit for information traveling between the set top box 108 and the headend 104 of the cable TV provider. In this embodiment, the network has one hundred and twenty analog channels and a bi-directional control data channel. Generally, the analog channels carry content and the control data channel carries control and entitlement information. Each analog carrier channel has a number of digital channels multiplexed into one data stream where the digital channels are distinguished by packet identifiers (PIDs). The bi-directional control channel is an outof-band channel that broadcasts data to the set top boxes 108 at one frequency and receives data from the boxes 108 at another frequency. Return data may be queued to decrease overloading during peak use periods using a store and forward methodology well known in the art. Other embodiments could use a cable modem, digital subscriber line (DSL), cellular data, satellite links, microwave links, or carrier current techniques for both control information and content where the content is formatted as packet switched data.

The printer 212 is an optional accessory some users may purchase and add to their display system 200. When using the set top box 108 for personal computer tasks, the printer 212 allows printing data such as email, web pages, billing information, etc. As will be explained further below, the ability to use a peripheral such as a printer is regulated by an authorization check. Using this regulation feature, printers 212 compatible with the set top box 108 do not work unless proper authorization is obtained to enable that printer 212 for that set top box 108.

The TV display 216 presents the user with audio, text and/or video corresponding to the content. The display 216 typically receives an analog video signal that is modulated on a carrier corresponding to channel three, channel four or a composite channel. The set top box 108 produces a NTSC signal, for example, modulated to the appropriate channel. Other embodiments could use a video monitor or digital display instead of a television display 216. Use of a digital display would alleviate the need for an analog conversion by the set top box 108 because digital displays, such as liquid crystal displays, use digital information to formulate the displayed picture.

The wireless input device 218 allows interaction between the user and the set top box 108. This device 218 could be a remote control, mouse, keyboard, game controller, pen tablet or other input mechanism. An infrared transceiver on the input device 218 communicates with a similar transceiver on the set top box 108 to allow wireless communication. In other embodiments, RF link or wired link could be used instead of the infrared transceiver.

The set top box 108 has component parts that perform authentication and authorization of objects and resources. Objects are any collection of digital information such as software, drivers, firmware, data, video, or audio. Resources are anything needed by an object to operate as intended such as another object or a physical device. Included in the set top box 108 are a controller 220, memory 228, a printer port 232, a network port 236, an access control processor 240, a display interface 244, and an infrared (IR) port

248. These blocks communicate with each other over a bus 230 where each block has a different address to uniquely identify it on the bus 230. Typically, the set top box 108 is a separate device, but could be integrated with the TV display 216, a computer, an information appliance, or personal video recorder (PVR).

The controller 220 manages operation of the set top box 108 using a trusted or secure operating system. Such functions as digital object decryption and decompression are performed in the controller 220 as well as functions such as switching TV channels for the user and presenting menus to the user. Included in the controller are a processor, an encryption engine, local memory, and other items common in computing systems.

In other embodiments, the controller 220 could also contain an adjunct secure microprocessor for purposes of key protection or cryptographic processing. This may be appropriate in some systems where a high level of security is desired.

The set top box 108 includes a block of memory 228. This memory 228 is solid state memory that could include RAM, ROM, flash, and other types of volatile and non-volatile memory. Objects and resources are stored in memory for running at a later time. During execution, programs are loaded into and executed within the memory 228, and also use the memory 228 for scratchpad space. Keys, serial numbers and authorizations can be stored in non-volatile flash memory.

This embodiment includes a printer port 232 for interfacing to an optional printer 212. The printer port 232 resource is not available to programs unless authorized. As explained further below, each object must have authorization to use a resource such as the printer port 232. Data is sent from the printer port 232 to the printer 212 in a serial or parallel fashion by way of a wired or wireless transport mechanism.

Stated generally, a checkpoint is a point in time or a step of processing where the authentication and/or authorization status of a functional unit is confirmed. A checkpoint is encountered when printing is requested. The checkpoint authorizes and authenticates the object requesting the printing. Checkpoints are places in one object where authentication and/or authorization are run on another object (e.g., an operating system checks authentication and authorization of an application that is running). Ideally, checkpoints are performed when the purpose of the object becomes manifest. In the case of a printer port 232, its purpose becomes manifest when it is used to print something. Accordingly, a checkpoint is triggered to check the object using the printer port 232 resource when anything is printed. Typically, the checkpoint for printing would be in the operating system.

The network port 236 allows bi-directional communication between the set top box 108 and the headend 104. Included in the network port 236 are a tuner and a demodulator that tune to analog carrier channels and demodulate an MPEG data stream to allow one-way delivery of content. Also included in the network port 236 is a control data transceiver or cable modem that allows for bi-directional communication of control data information and/or content. To distribute loading of the control data path to the headend 104 more evenly, a store and forward methodology may be used.

Modulation of the digital video signal onto an analog signal compatible with the TV display 216 is performed by the display interface 244. As discussed above, the TV display 216 generally accepts signals modulated on channel three, channel four or a composite channel. For displays that accept a digital input, such as LCD displays, the display interface 244 performs any formatting required by the digital input.

The IR port 248 communicates bi-directionally with a wireless input device 218. Included in the IR port 248 is an IR transceiver that provides the wireless communication path with the input device 218. Other electronics in the IR port 248 convert analog signals received by the transceiver to a corresponding digital signal and convert analog signals sent to the transceiver from a corresponding digital signal. The controller 220 processed the digital signals so that the user can control some of the functions within the set top box 108.

The access control processor (ACP) 240 regulates security functions within the set top box 108. For example, the ACP 240 performs authentication and authorization either under the direction of the controller 220 or independent of the controller 220 as will become clear in the discussion below. To perform its tasks, the ACP 240 includes a processor, RAM and ROM that cooperate to execute software independent of the controller 220. The ACP 240 also includes a decryption engine and a hash function for deciphering content and calculating signatures. Checkpoints are embedded into the software run that trigger the ACP 240 to perform security checks. In this embodiment, the ACP 240 is implemented in hardware, but other embodiments could perform the functions of the ACP 240 in software.

The ACP 240 can also shadow the operating system (OS) to assure proper functioning of the OS. By watching the launch of objects, the ACP 240 can monitor which application objects are running. If necessary, the ACP 240 can kill running applications if a checkpoint detects an error or if authorization expires. Further, the ACP 240 could monitor memory 228 to detect any application not authorized to be in memory 228. Scratchpad memory size could also be monitored to detect applications hiding in scratchpad memory. Additionally, the ACP 240 could randomly execute checkpoints on the objects in memory to confirm their authorization and/or authenticity. Problems encountered by the ACP 240 are reported to either the OS or the headend 104. In these ways, the ACP 240 acts as a software security guard bot within the set top box 108 such that aberrant behavior is detected and reported.

With reference to Figs. 3, an embodiment of an object message 300 is shown in block diagram form. Object messages 300 deliver functional units to the set top box 108 from the network 208 such that information may be sent to the set top boxes 108 after they are fielded. Forming the object message 300 are an object header 304, an object 308 and requirements 312. This embodiment includes requirement information in the object message 300, however, other embodiments could use a separate "requirements" message to convey this information. Although not shown in Fig. 3, checksums or digital signatures are used to validate that the object message 300 is transported to the set top box 108 from the headend 104 without errors.

The object header 304 includes attributes for the object message 300. Included in the object header 304 are a header length, an object length, a functional unit identifier, a software version, and a domain identifier. The header and object lengths respectively indicate the lengths of the object header 304 and the object 308. For authentication purposes, among other reasons, the functional unit identifier provides a unique code that allows attributing a "rights" message to the object message 300. The software version indicates the revision number of the object. Different cable TV providers are assigned domain identifiers such that all of the set top boxes 108, which might receive an object 308, can screen for objects 308 associated with their domain.

The object 308 includes content the system 100 is designed to deliver to set top boxes 108. Upon download of the object 308, it is authenticated and authorized to verify the source of the object message 300 and availability of the object 308 to the receiving set top box 108. Several types of content or information can be embedded in an object, such as executable programs, firmware upgrades, run-time programs (e.g., Java® or ActiveX®), programming schedules, billing information, video, audio, and/or data. The object 308 can be used immediately after authentication and authorization or at a later time. Additionally, authorization can be programmed to expire after a certain amount of time or can be rechecked periodically as the object 308 is used.

The requirements data structure 312 allows the content provider to limit access of the object 308 to predetermined subset of all the set top boxes 108. Each functional unit identifier is mapped to one or more rights. Rights are transported in "rights" messages to enable operation of a functional unit that is mapped to those rights in the requirements data structure 312.

Referring next to Fig. 4, an embodiment of a "rights" message 400 is shown in block diagram form. The rights message 400 conveys rights to use a functional unit. The functional unit could be an object or a resource. Typically, there is one rights message 400 for each set top box 108, which specifies any rights for all functional units. Requirements associated with the objects and resources are checked against the rights to determine if interaction with another the object or resource is authorized. The rights message 400 allows remotely adding new rights to a functional unit associated with the set top box 108. Although not shown, the rights message 400 typically includes a digital signature to verify the integrity of the message 400 during transport. In some embodiments, a checksum could be used instead of a digital signature.

The rights header 404 includes attributes for the rights message 400. Included in the rights header 404 are a header length, a rights data structure length, a functional unit identifier, and a domain identifier. The header length and the rights data structure length respectively indicate the lengths of the rights header 404 and the rights data structure 408. For authentication purposes, the functional unit identifier provides a unique code that allows attributing the rights message 400 to a particular functional unit.

Rights are conveyed to the functional units using the information in the rights data structure 408. A given functional unit may have rights to use several other functional units. These rights are contained in the rights data structure 408. The functional unit identifier in the rights header 404 is used to attribute the rights to a particular functional unit. The functional unit may be already in the set top box 108 or may be downloaded at some later time.

Referring next to Fig. 5, an embodiment of a "requirements" message 500 is shown in block diagram form. The requirements message 500 is used for resources already resident in the set top box 108. Included in the requirements message 500 are a requirements header 504 and a requirements data structure 508. The requirements message 500 is similar to the object message 300 except that there is no embedded object

308 as it is already located in the set top box 108. Although not shown, the requirements message 500 typically includes a digital signature to verify the integrity of the message 500 during transport.

The requirements header 504 includes attributes for the requirements message 500. Included in the requirements header 304 are a header length, a requirements data structure length, and a domain identifier. The header and requirements data structure lengths respectively indicate the lengths of the requirements header 504 and the requirements data structure 308. Different cable TV providers are assigned domain identifiers such that all of the set top boxes 108, which might receive an object 308, can screen for objects 308 associated with their domain.

The requirements data structure 508 allows the content provider to limit access of a resource to predetermined subset of all set top boxes 108. Each functional unit identifier is mapped to one or more requirements by the requirements data structure 508. Rights are transported in the rights message 400 to enable operation of a functional unit so long as the rights in the rights message 400 satisfy the requirements in the requirements data structure 508 for that functional unit.

The object message 300 is uniquely coupled with the associated requirements message 500 by a signature over both messages 300, 500. Even though transported separately, the common signature assures the pair of messages 300, 500 are not modified during transport.

With reference to Fig. 6, some of the functional units of a set top box 108 are shown. Functional units toward the bottom of Fig. 6 are superordinate to the functional units near the top of Fig. 6. That is to say, functional units toward the top of Fig. 6 are subordinate to those lower in the figure. Superordinate functional units are responsible for imposing checkpoints on subordinate functional units. For example, the hardware 604 imposes checkpoints upon the BIOS 608, OS 612 and so on up the subordination hierarchy. The BIOS 608 imposes checkpoints on the OS 612, but not upon the hardware 604. Functional units in the same ordination stratum can impose a checkpoint on another functional unit in that stratum when they interact. For example, an application 616 can require execution of a checkpoint on a driver 618.

Superordinate functional units are designed to initiate execution of the checkpoints in conjunction with the ACP 240 and subordinate objects are designed to have checkpoints imposed upon them. For example, the BIOS 608 requires execution of a checkpoint upon the OS 612 during the boot process, during execution and/or periodically while running. A driver object 618 is subject to checkpoints when installed or exercised during normal operation. Data file objects 622 are subject to checkpoints whenever the data in the file is accessed. An HTML object 628 is reviewed as part of a checkpoint whenever the HTML object 628 is interpreted by a browser application 616.

Referring next to Fig. 7, interaction between functional units is shown in block diagram form. The functional units associated with the set top box 108 include a set top box resource 704, a printer driver object 708, an e-mail object 712, and a printer port resource 716. During the normal interaction of these functional units, checkpoints are encountered that trigger authorization checks. The sole table correlates rights and requirements to each functional unit in Fig. 7. The functional unit identifier serves to correlate the object messages 300 with the rights messages 400.

Table

Functional Unit ID	Functional Unit	Requirements	Rights
604	Set Top Box	NA	E-mail, Printer Driver, etc.
612	E-mail	Yes	Printer Driver
608	Printer Driver	Yes	Printer Port
614	Printer Port	Yes	None

The set top box resource 704 is superordinate to the email object 712. When the email object 712 is loaded, a checkpoint in the object 712 checks for proper rights. The proper rights are defined by the requirements 720-2 of the email object 712 itself. If the e-mail right 716-1 meets the standards of the e-mail object requirements 720-2, the e-mail object 712 continues execution past the checkpoint. The ACP 240 actually performs the authentication after the e-mail right 716-1 and e-mail object requirements 720-2 are respectively loaded by their associated functional units 704, 712.

After the user receives the set top box 704, the user can add an optional printer 212. In this embodiment, the ability to print is an added feature that is not included in all set top boxes 704. If the printer 212 is a purchase sanctioned by the content provider, printer driver rights 716-2, 716-4 and a printer port right 716-3 are sent in rights messages 400 to the set top box 704 from the headend 104.

Some embodiments could provide rights to a subset of the functional units capable of using the printer port 720-3. For example, the e-mail object 712 could be given the printer driver right 716-4, but the set top box resource 704 would not receive the printer driver right 716-2. In this way, only the email object 716-2 could use the printer port 720-3 and the other objects could not.

Hooking the printer to the to the printer port can trigger display of a message on the TV 216 that asks for a secret code included with the printer 212. After the user enters the secret code, a request for the rights messages 400 that enable the printer is made to the headend 104. Once the headend receives and verifies the secret code, an enabling set of rights messages 400 are sent encrypted in a key based upon the secret code. In this embodiment, the printer driver object 708 is factory loaded, but other embodiments could load this object 708 when needed using an object message 300.

While the e-mail object 712 is running, the user may try to print an e-mail message. Several checkpoints authenticate the proper rights 716 are present before printing. The e-mail object 712 calls the printer driver 708 with the information requiring printing. A checkpoint in the printer driver 708 stops processing until the authorization of the e-mail object 712 is checked. A printer driver right 716-4, downloaded when the printer was purchased, is loaded into the ACP 240 along with the printer driver requirements 720-1 for authentication. Presuming authentication is successful, the printer driver object 708 formats the print information for the printer 212 and passes it to the printer port resource 714.

The printer port resource 714 is the hardware port that interfaces to a cable connected to the printer 212. Once information is sent to the printer port resource 714 a checkpoint pauses the processes to check that the printer driver object 708 has proper authorization. The requirements 720-3 and rights 716-3 are loaded into the ACP 240 for authentication. Once the use by the printer driver object 708 is authenticated, the remainder of the print job is spooled to the printer port resource 714 for printing.

In some embodiments, the rights 716 of one functional unit can be inherited by another functional unit. The right 716 could be conveyed to other objects 308 that might use that functional unit. For example, the right 716 to use the printer port 232 could initially be associated with the e-mail object 712 alone, where this right 716 is conveyed to e-mail object 712 when the user purchased a printer 212. At a later time, the headend 104 could authorize inheritance of that right 712 to all other functional units or subset of the functional units that might use the printer port 232. In this way, additional functional units could use the print feature.

With reference to Fig. 8, a flow diagram of an embodiment of a process for loading functional units is illustrated. This embodiment allows factory loading or field loading of functional units. The process begins in step 804 where the functional units are designed. Functional units can be objects and/or resources. The resources include hardware such as the set top box 108 and the hardware components inside the set top box 108. During step 804, the various requirements for the functional units are defined.

A determination is made in step 808 as to whether the functional unit is being installed in the factory or in the field. As those skilled in the art appreciate, resources that are physical devices are typically installed at the factory. Electronically storable objects can generally be installed in the factory or the field, however, certain objects are installed in the factory such as portions of the operating system 612.

If the particular functional unit being installed is factory loaded, processing continues to step 812 where the functional units are installed into the set top box enclosure. Typically, the physically devices and most of the objects are factory installed such that the set top box is functional before shipment to the user. Certain objects 308, however, are loaded into the set top box 108 after fielding it.

For field loaded objects, processing goes from step 808 to step 820 where the objects are distributed to content providers. The distribution process includes electronically sending the object 308 by some type of data link such as a packet switched network. The content provider embeds the objects 308 in object messages 300 and broadcasts the objects 308 to set top boxes 108 over an entitled channel in step 824. The entitlement process for the channel includes sending keys to the entitled set top boxes such that they can decode the data stream on the channel. In step 828, the rights 716 for running the objects are determined by the content provider. The rights messages 400 are distributed in step 832. Only a subset of the set top boxes 108 who are broadcast the object 308 are typically allowed to decode the corresponding rights message 400. In this embodiment, the object message 300 is sent before the rights message 400, but other embodiments could reverse the order of sending the messages 300, 400.

Referring next to Fig. 9, an embodiment of a process for receiving an object 308 is shown in flow diagram form. This embodiment receives the object and rights messages 300, 400 before authenticating their use. In steps 904 and 908, the object and rights messages 300, 400 are received from the cable TV provider by way of an entitled channel. Entitlement of a channel involves encryption of the channel for one or more set top boxes 108 using symmetric or asymmetric encryption techniques. Signatures can also be used to verify the source of the messages 300, 400 is authentic and verify their authorization for use on the set top box 108. Other embodiments, could receive the rights message 400 before the object message 300.

Once successfully the messages 300, 400 are received and decrypted, they are correlated to each other using the functional unit identifiers in step 916. In step 920, the requirements 720 embedded in the object 308 are checked against the rights 716 in the rights message 400. This check of rights 716 against requirements 720 allows a determination as to whether the object 308 is authorized in step 924. If the object 308 is not authorized, an error is reported back to the headend 104 and the object 308 is discarded in steps 928 and 932. Alternatively, use of the object 308 is allowed in step 936 if the rights 716 satisfy the requirements 720. In this way, the object 308 is sent to the set top box 108 and checked for authorization.

Although the forgoing embodiment downloads an object 308 and authorizes it, other embodiments could authorize resources already in the set top box 108 without requiring download of the resource. Under this scenario, the rights message 400 is checked against the requirements already stored in the set top box 108.

With reference to Fig. 10, a flow diagram illustrating an embodiment of a process for authenticating and authorizing a software object 308 is shown. The process begins in step 1004 where the software object 308 is read into the controller 220 from memory 228. A signature associated with the software object 308 is checked in step 1008 to determine if the object 308 is authentic. As mentioned above, the signature could be generated over both the object message 300 and requirements message 500 to provide authentication of both messages 300, 500 with the single signature. The software object 308 and requirements message 500 are loaded into the ACP 240 to calculate the signature. The calculated signature is checked against the original signature sent from the headend 104 to check authentication.

If it is determined that the software object fails the authentication as determined in step 1012, processing continues to steps 1016 and 1020. In step 1016, an error is reported to the headend 104. This may happen immediately where a bi-directional control data channel is available or may happen at a later time where a store-and-forward methodology is used. Objects 308 that fail the authenticity test are corrupt and are discarded in step 1020. A message may or may not be presented to the user after failure of authentication.

In step 1032, a check of authorization is performed for a resource to use an object 308. The rights 716 of the set top box 108 are checked against the requirements

720 for the resource. It is noted, the resource could be the same object 308 being executed such that a checkpoint causes an authorization check of the very object 308 that includes the checkpoint. The ACP 240 is involved in this authorization process to securely check rights against requirements..

A determination is made in step 1036 as to whether the interaction between object 308 and resource is authorized. If the use is not authorized, the error is reported to the user and/or headend 108 in step 1040. For example, a message could appear on the screen to notify the user that the printer port is unavailable and that the content provider should be contacted to enable this feature. If the authorization check is successful, the object 308 is allowed to begin execution in step 1024.

In another embodiment, authorization could be checked before authentication rather than the other way around. Authorization is typically a quicker check than authentication and a negative result from an authorization check could avoid unnecessarily checking authentication.

In yet another embodiment, the resource accessed by the object 308 could also encounter a checkpoint that requires authentication. For example, the printer port resource 232 may have a requirement 720 that needs satisfaction by a corresponding right 716. The ACP 240 is once again used to perform this authentication. In this way, various functional units have checkpoints applied to them to check authentication and/or authorization.

With reference to Fig. 11, a flow diagram illustrating an embodiment of a process for authenticating the running of a software object and for authorizing that software object 308 is shown. The process begins in step 1104 where the software object 308 is read into the controller 220 from memory 228. The software object 308 is checked in step 1108 to determine if the object 308 is authentic and authorized.

If it is determined that the software object fails either the authorization or authentication as determined in step 1112, processing continues to steps 1116 and 1120. In step 1116, an error is reported to the headend 114. This may happen immediately where a bi-directional control data channel is available or may happen at a later time where a store-and-forward methodology is used. Objects 308 that fail the authentication or authorization test are corrupt and are discarded in step 1120.. A message may or may not be presented to the user after a failure.

Presuming it is determined that the software object 308 is authorized and authentic in step 1112, execution of the software object 308 is begun in step 1124.

During execution, a checkpoint is encountered in the software object 308 that requires authorization in step 1128. Checkpoints are typically added to the software object 308 in places where the purpose of that object 308 becomes manifest. For example, a printer driver would have a checkpoint before the driver is used to access the printer port 232. The checkpoint could require authorization and/or authentication of the object containing the checkpoint and/or another resource.

In step 1132, a check of resource authorization is performed for an object 308. The resource rights 716 of the set top box 118 are checked against the resource requirements 720 for the object. It is noted, the resource needed could be the same object 308 being executed such that the checkpoint causes an authorization check of the very object 308 that includes the checkpoint. The ACP 240 is involved in this authorization process to securely check rights against requirements.

A determination is made in step 1136 as to whether the interaction between object 308 and the necessary resource is authorized. If the use is not authorized, the error is reported to the user and/or headend 118 in step 1140. For example, a message could appear on the screen to notify the user that the printer port is unavailable and that the content provider should be contacted to enable this feature. If the authorization check is successful, the object 308 is allowed to use the resource in step 1144.

There are many permutations of checking authorization and authentication of software objects and authorization of the resources the software objects access. Only some of those permutations are described herein, but it is to be understood the invention includes these permutations. The above embodiment checks authorization and authentication of the software object before execution and checks authorization of the resource at a checkpoint during execution. Other embodiments, however, could perform the authorization of the resource before execution of the software object and the checkpoint could perform authorization and authentication of the software object again.

With reference to Fig. 12, a flow diagram illustrating another embodiment of a process for authenticating the running of a software object and for authorizing that software object 308 is shown. The process begins in step 1204 where the software object 308 is read into the controller 220 from memory 228. A signature associated with the software object 308 is checked in step 1208 to determine if the object 308 is authentic. As mentioned above, the signature could be generated over both the object message 300 and requirements rights message 400 500 to provide authentication of both messages 300, 400 500 with the single signature. The software object 308 and rights requirements message 400 500 are loaded into the ACP 240 to calculate the signature. The calculated signature is checked against the original signature sent from the headend 124.

If it is determined that the software object fails the authorization as determined in step 1212, processing continues to steps 1216 and 1220. In step 1216, an error is reported to the headend 124. This may happen immediately where a bi-directional control data channel is available or may happen at a later time where a store-and-forward methodology is used. Objects 308 that fail the authenticity test are corrupt and are discarded in step 1220. A message may or may not be presented to the user after failure of authorization.

Presuming it is determined that the software object 308 is authentic in step 1212, execution of the software object 308 is begun in step 1224. During execution, a checkpoint is encountered in the software object 308 that requires authorization in step 1228. Checkpoints are typically added to the software object 308 in places where the purpose of that object 308 becomes manifest. For example, a printer driver would have a checkpoint before the driver is used to access the printer port 232. The checkpoint could require authorization and/or authentication of the object containing the checkpoint and/or another resource.

In step 1232, a check of authorization is performed for an object 308 to use a resource. The resource rights 716 of the set top box 128 are checked against the requirements 720 for the object 308. It is noted, the resource could be the same object 308 being executed such that the checkpoint causes an authorization check of the very object 308 that includes the checkpoint. The ACP 240 is involved in this authorization process to securely check rights against requirements.

A determination is made in step 1236 as to whether the interaction between object 308 and resource is authorized. If the use is not authorized, the error is reported to the user and/or headend 128 in step 1240. For example, a message could appear on the screen to notify the user that the printer port is unavailable and that the content provider should be contacted to enable this feature. If the authorization check is successful, the object 308 is allowed to use the resource in step 1244.

In light of the above description, a number of advantages of the present invention are readily apparent. Functional units such as object or resources can be controlled in a computer system at any time a checkpoint is encountered by checking authentication and/or authorization. This ability allows conditional access and verification of functional units at multiple times during their use in a system. As those skilled in the art can appreciate, these techniques reduce the threat of viruses and other unwanted modifications or substitutions of functional units.

A number of variations and modifications of the invention can also be used. Some embodiments could use the ACP to perform authentication and authorization checks while others could perform these checks with similar software algorithms.

Although the invention is described with reference to specific embodiments thereof, the embodiments are merely illustrative, and not limiting, of the invention, the scope of which is to be determined solely by the appended claims.

WHAT IS CLAIMED IS:

1. A method for securing information in a conditional access system, the method comprising:

providing a functional unit;

providing requirements related to the functional unit;

entitling a transmission conduit to a content receiver;

receiving rights related to the functional unit;

correlating the rights with the requirements; and

checking the rights against the requirements.

2. The method for securing information in the conditional access system of claim 1, wherein the functional unit includes at least one of an object and a resource.

3. The method for securing information in the conditional access system of claim 2, wherein the object comprises at least one of software, drivers, firmware, data, video, and audio.

4. The method for securing information in the conditional access system of claim 2, wherein the resource comprises at least one of an object and a physical device.

5. The method for securing information in the conditional access system of claim 1, further comprising authenticating a source of the functional unit.

6. The method for securing information in the conditional access system of claim 1, further comprising authorizing use of the functional unit if the right satisfies the requirement.

7. The method for securing information in the conditional access system of claim 1, wherein the entitling the transmission conduit comprises decrypting information passing through the transmission conduit.

8. A method for securing information in a conditional access system, the method comprising:

entitling a transmission conduit to a content receiver;

sending a functional unit to the content receiver;

determining requirements for the functional unit; and

sending the requirements through the transmission conduit, wherein the requirements are related to the functional unit.

9. The method for securing information in the conditional access system of claim 8, comprising determining rights associated with the content receiver.

10. The method for securing information in the conditional access system of claim 8, allowing inheritance of a right from a first functional unit to a second functional unit.

11. The method for securing information in the conditional access system of claim 8, further comprising sending rights to the functional unit.

12. The method for securing information in the conditional access system of claim 11, further comprising:

correlating the rights to the functional unit; and

checking the rights against the requirements.

13. The method for securing information in the conditional access system of claim 11, wherein the entitling a transmission conduit comprises encrypting information passing to the transmission conduit.

14. The method for securing information in the conditional access system of claim 8, further comprising authenticating the functional unit.

15. A content receiver for a conditional access system, comprising:

a body;

an encrypted channel from a content provider to the body;

a functional unit within the body;

requirements related to the functional unit; and

rights related to the functional unit, wherein the rights are checked against the requirements to authorize use of the functional unit.

16. The content receiver for the conditional access system of claim 15, wherein the functional unit includes at least one of an object and a resource.

17. The content receiver for the conditional access system of claim 15, further comprising a second functional unit, wherein the rights of the functional unit are inherited by the second functional unit.

18. The content receiver for the conditional access system of claim 15, wherein a source of the functional unit is authenticated.

19. A method for authorizing interaction between functional units in a computing system, the method comprising:

providing a first functional unit associated with a requirement;

providing a second functional unit associated with a right;

initiating interaction between the first and second functional units;

checking the right against the requirement in response to the initiation of interaction between the first and second functional units; and

authorizing use of the first functional unit by the second functional unit.

20. The method for authorizing interaction between functional units in the computing system of claim 19, further comprising authenticating a source of at least one of the first functional unit and the second functional unit.

21. The method for authorizing interaction between functional units in the computing system of claim 19, further comprising entitling a transmission conduit for transporting at least one of the first functional unit and the second functional unit.

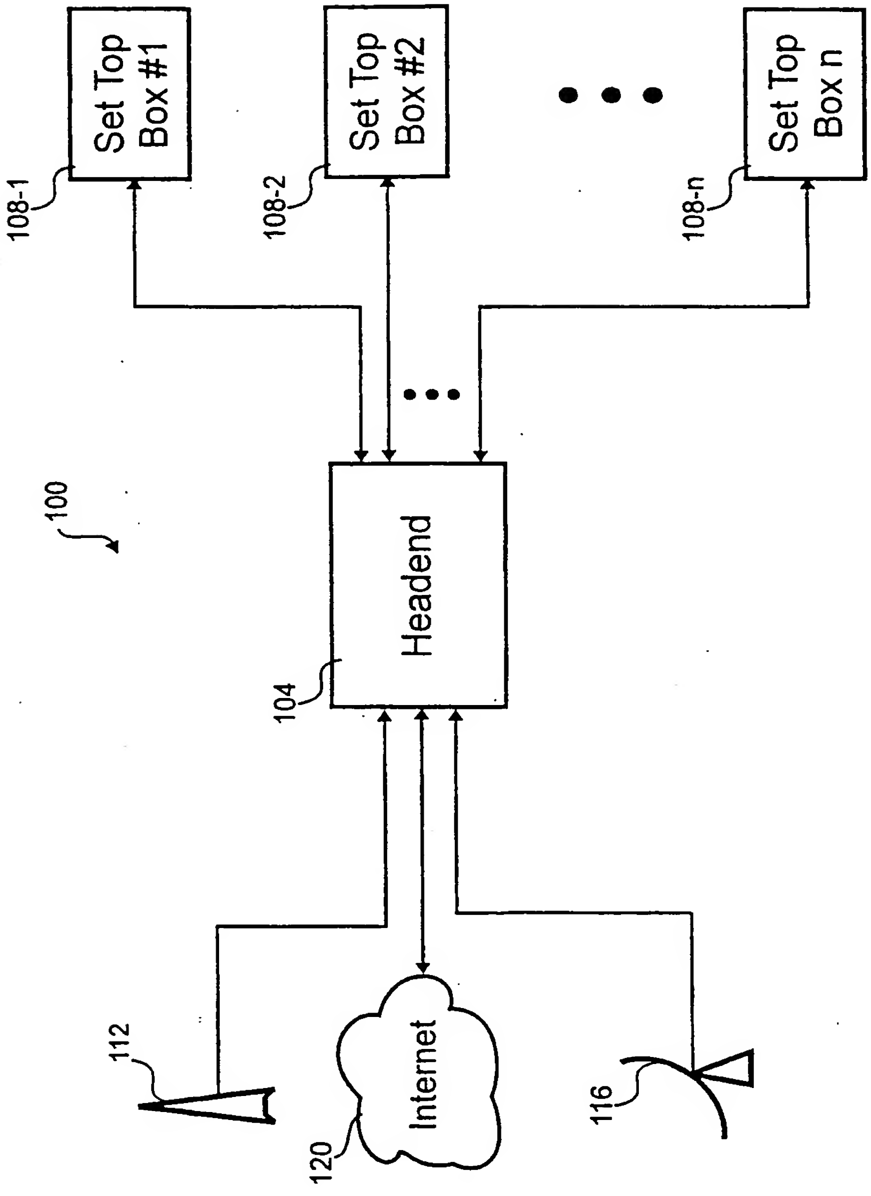

100
Set Top Box #1
108-1
Set Top Box #2
108-2
Set Top Box n
108-n
Headend
104
112
Internet
120
116

Fig. 1

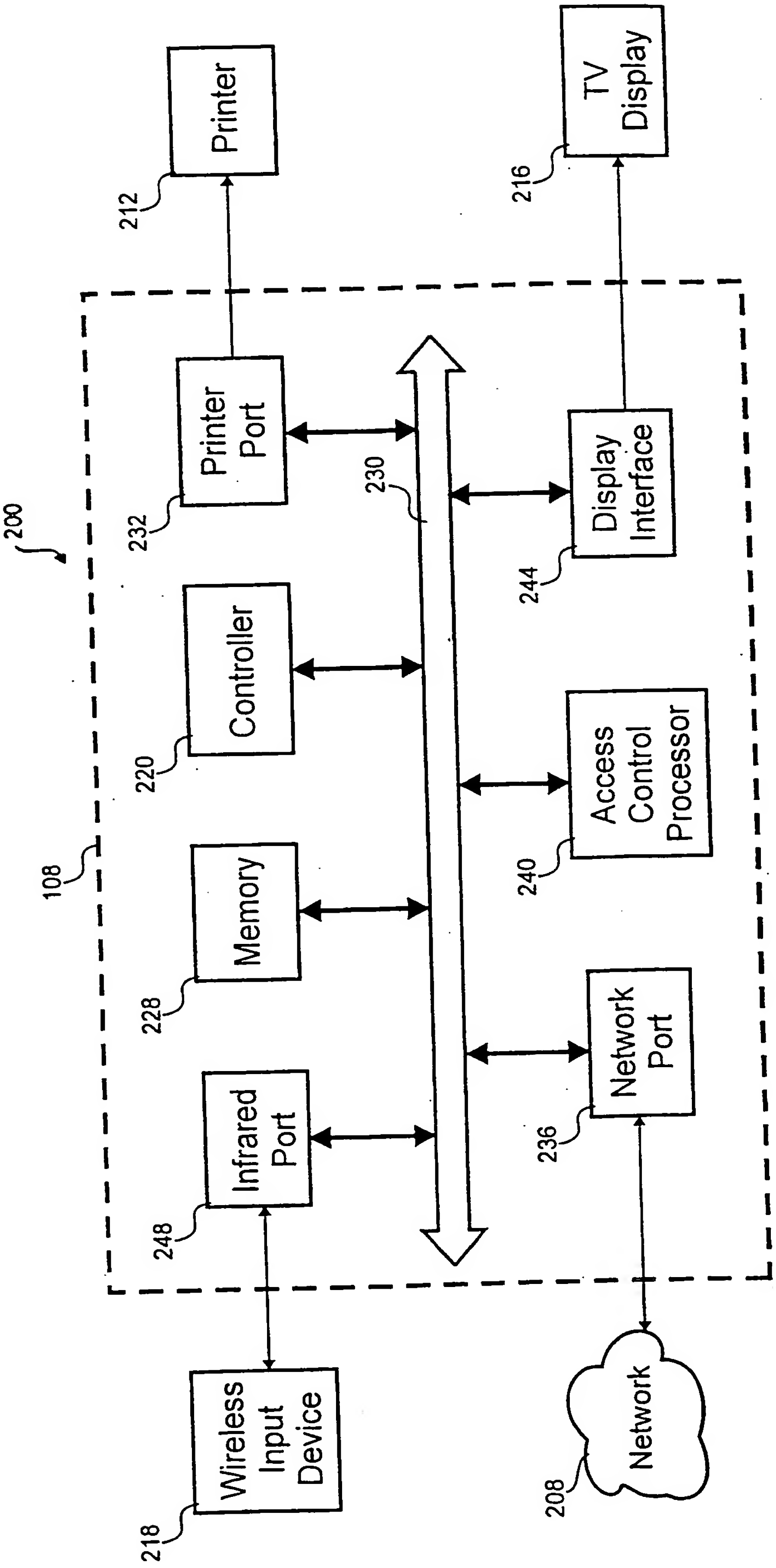

200
108
Printer
212
TV Display
216
Printer Port
232
Controller
220
Memory
228
Infrared Port
248
230
Display Interface
244
Access Control Processor
240
Network Port
236
Wireless Input Device
218
Network
208

Fig. 2

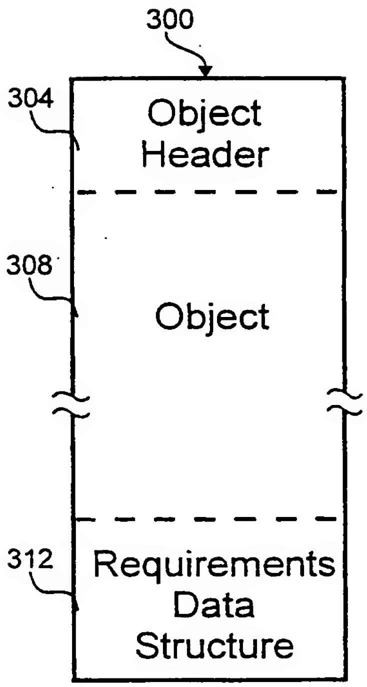


Fig. 3

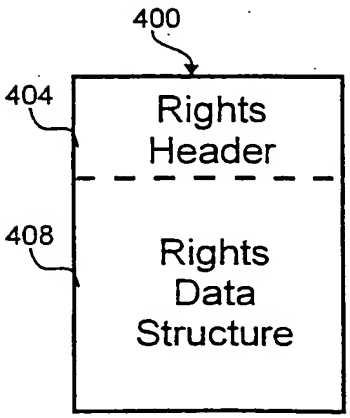


Fig. 4

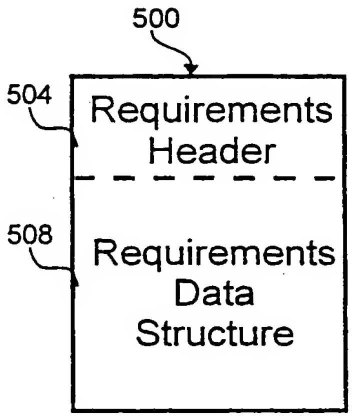


Fig. 5

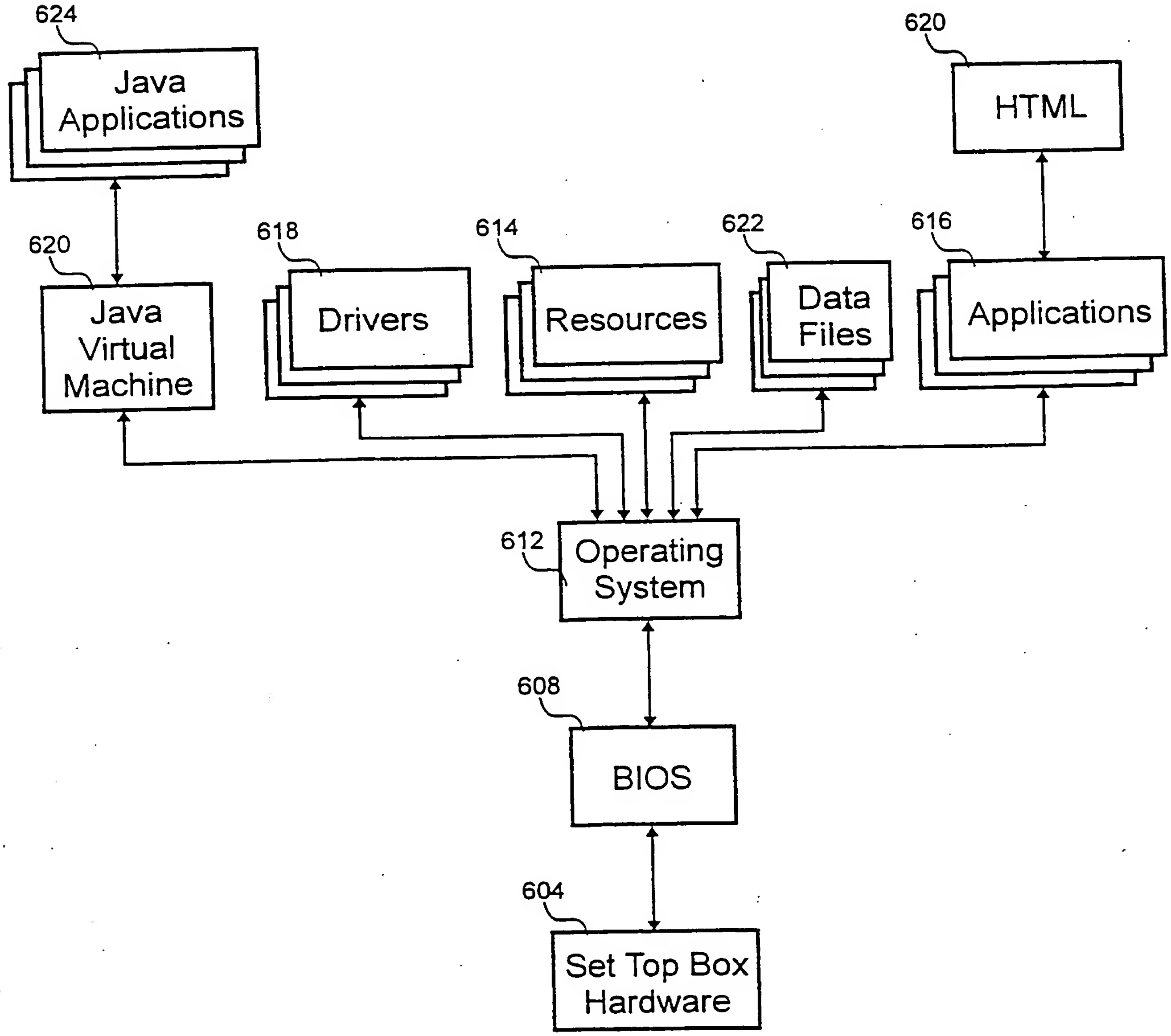

624
Java Applications
620
HTML
620
Java Virtual Machine
618
Drivers
614
Resources
622
Data Files
616
Applications
612
Operating System
608
BIOS
604
Set Top Box Hardware

Fig. 6

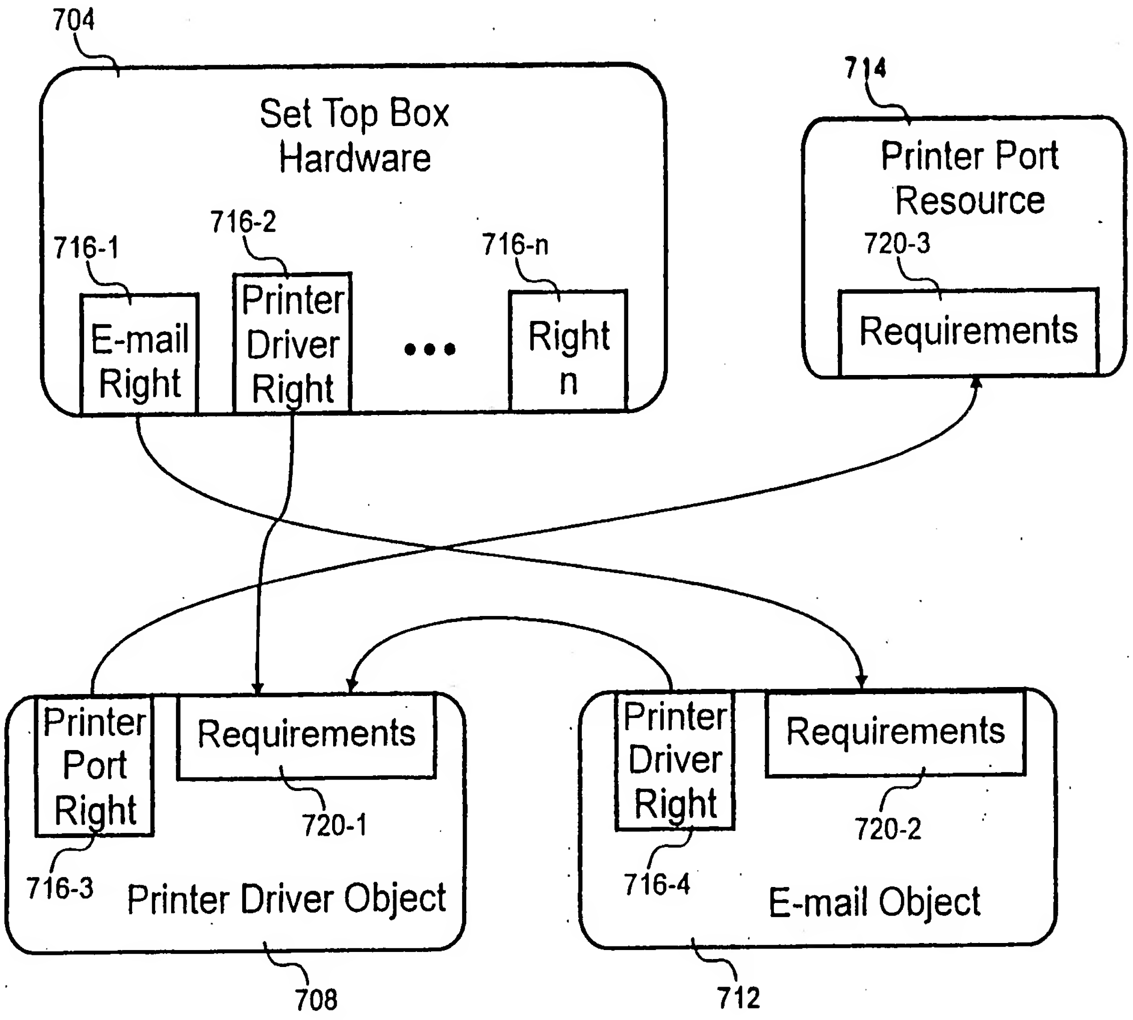

704
Set Top Box
Hardware
716-1
E-mail
Right
716-2
Printer
Driver
Right
•••
716-n
Right
n
714
Printer Port
Resource
720-3
Requirements
Printer
Port
Right
716-3
Requirements
720-1
Printer Driver Object
708
Printer
Driver
Right
716-4
Requirements
720-2
E-mail Object
712

Fig. 7

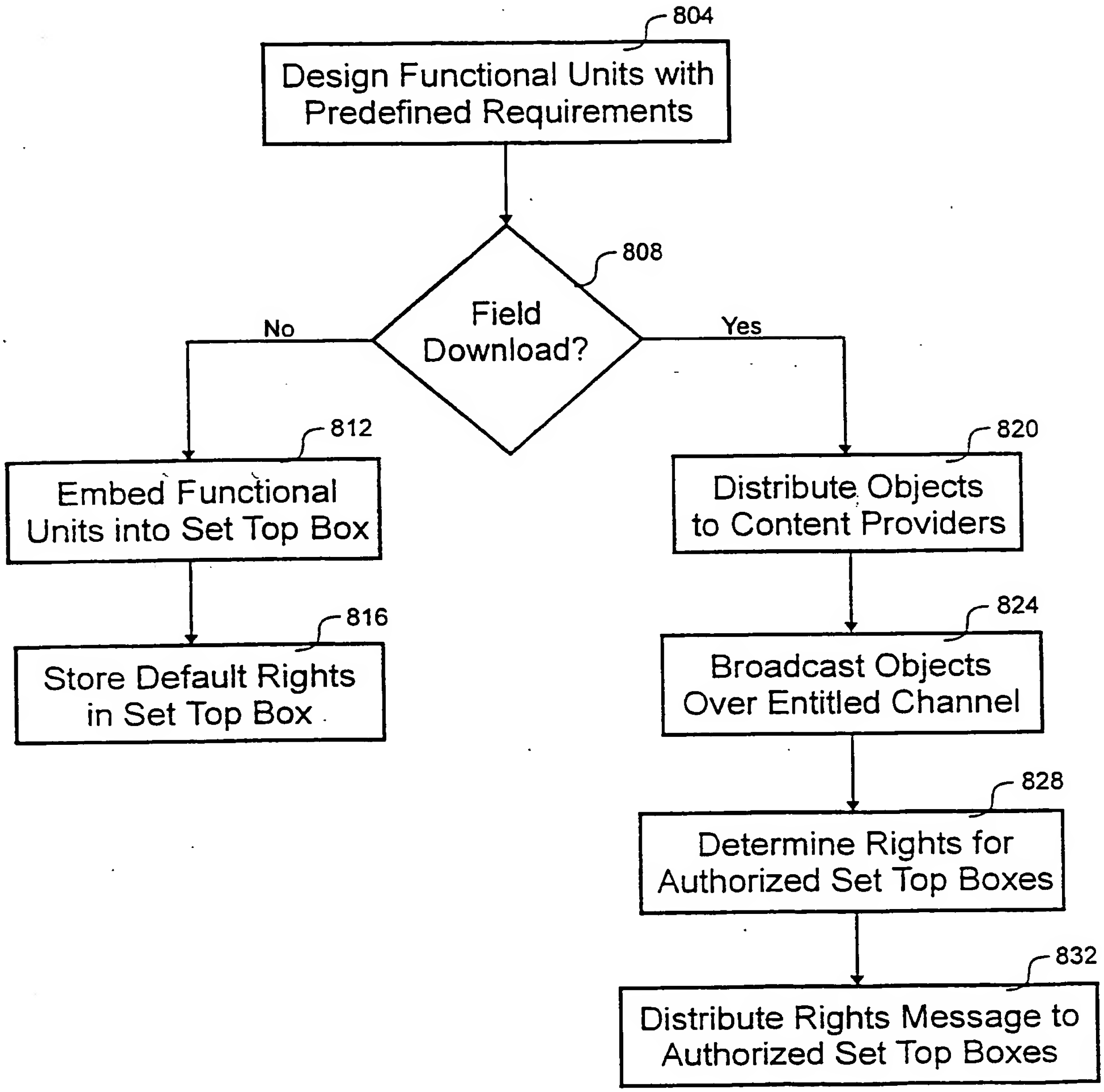

804
Design Functional Units with Predefined Requirements
808
Field Download?
No
Yes
812
Embed Functional Units into Set Top Box
816
Store Default Rights in Set Top Box
820
Distribute Objects to Content Providers
824
Broadcast Objects Over Entitled Channel
828
Determine Rights for Authorized Set Top Boxes
832
Distribute Rights Message to Authorized Set Top Boxes

Fig. 8

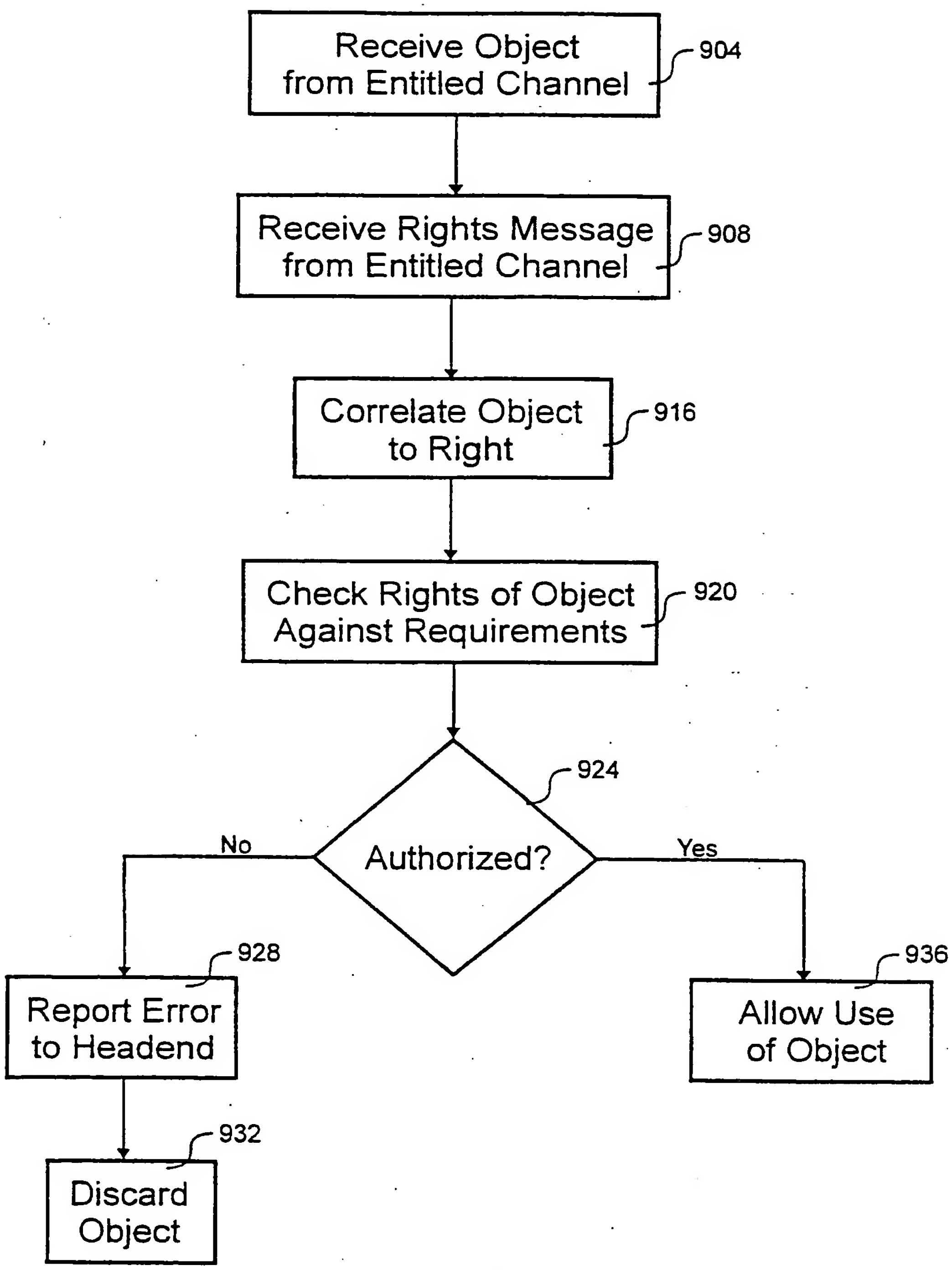

Receive Object from Entitled Channel
904
Receive Rights Message from Entitled Channel
908
Correlate Object to Right
916
Check Rights of Object Against Requirements
920
Authorized?
924
No
Yes
Report Error to Headend
928
Discard Object
932
Allow Use of Object
936

Fig. 9

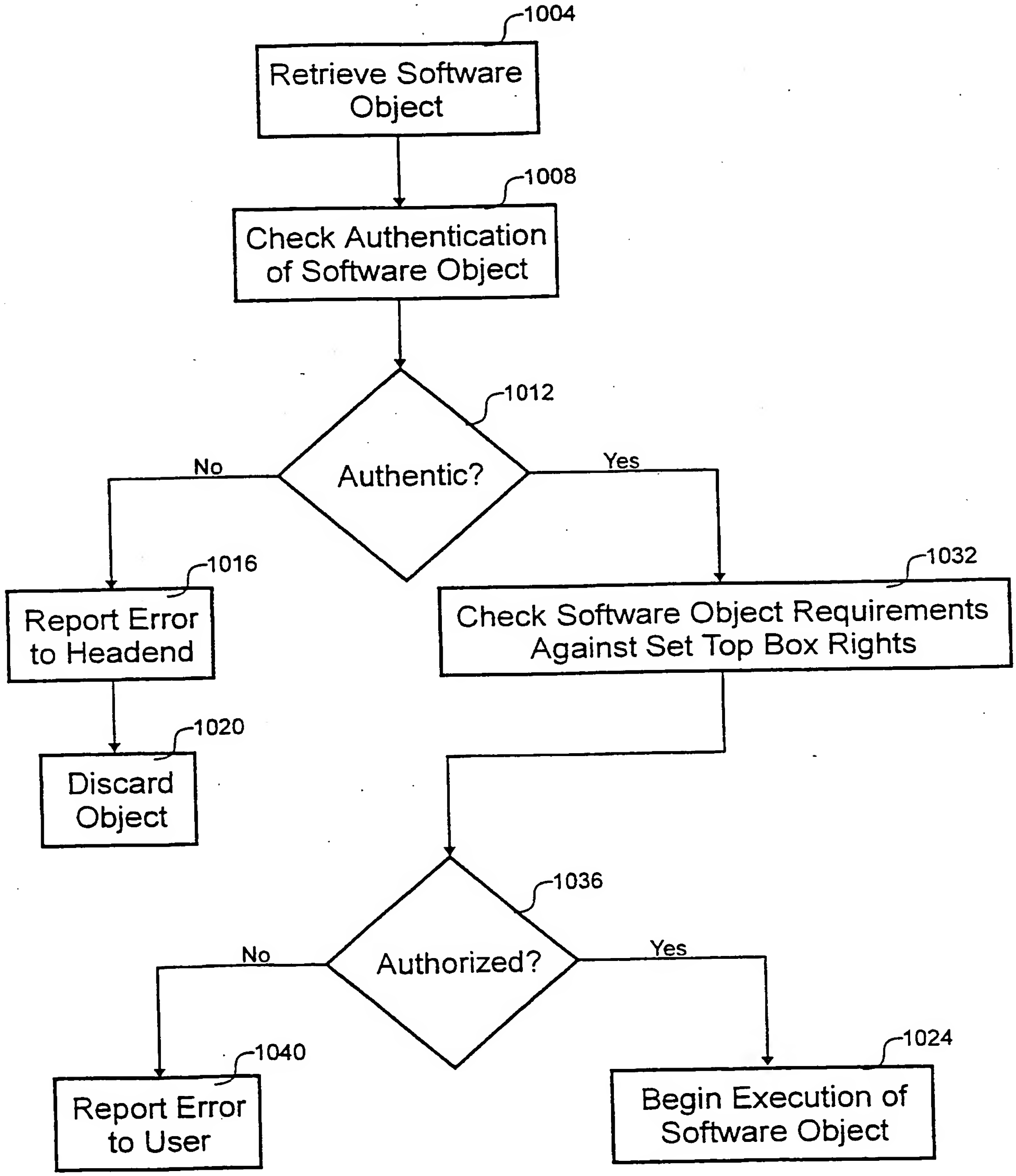

1004
Retrieve Software Object
1008
Check Authentication of Software Object
1012
Authentic?
No
Yes
1016
Report Error to Headend
1020
Discard Object
1032
Check Software Object Requirements Against Set Top Box Rights
1036
Authorized?
No
Yes
1040
Report Error to User
1024
Begin Execution of Software Object

Fig. 10

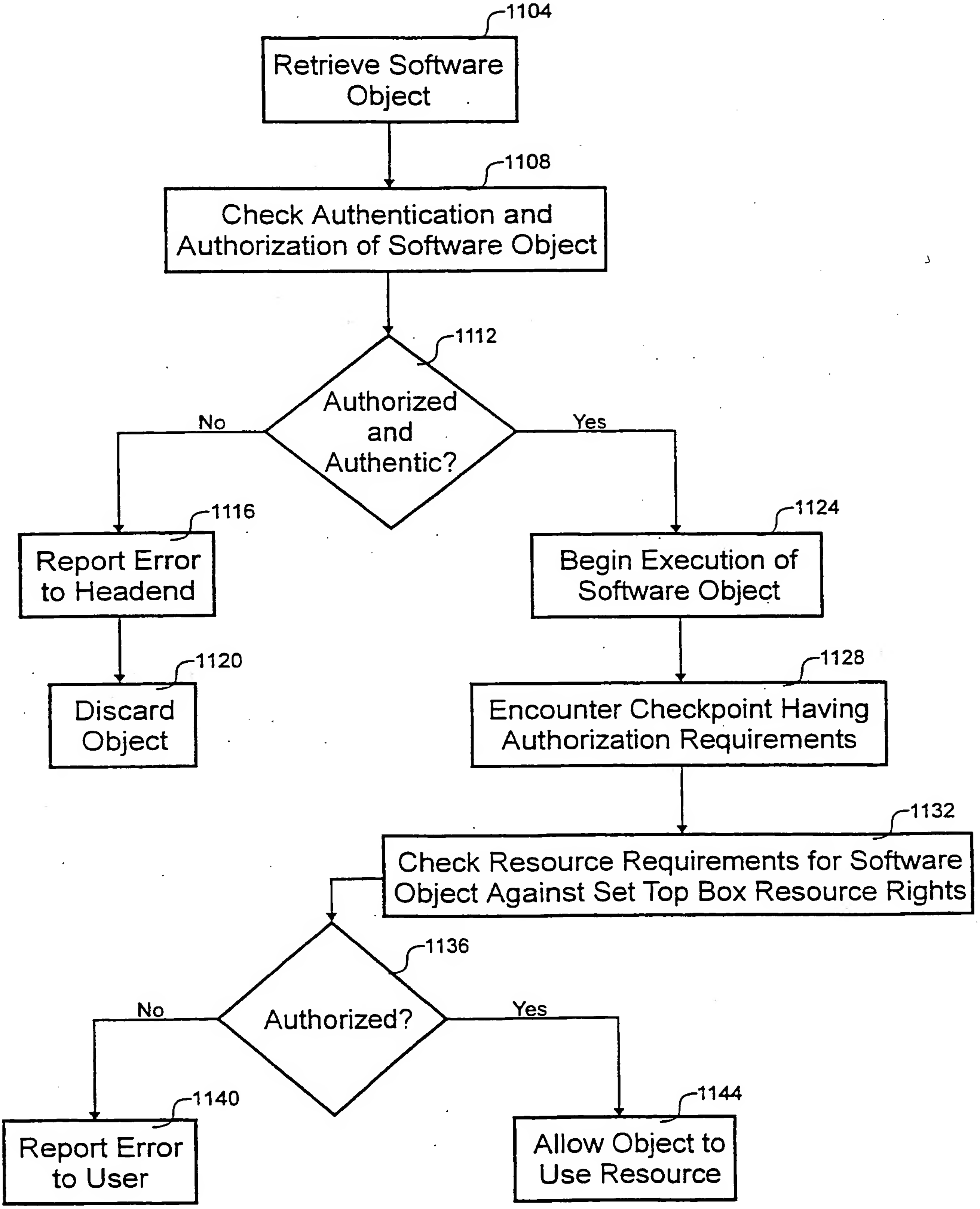

1104
Retrieve Software Object
1108
Check Authentication and Authorization of Software Object
1112
Authorized and Authentic?
No
Yes
1116
Report Error to Headend
1120
Discard Object
1124
Begin Execution of Software Object
1128
Encounter Checkpoint Having Authorization Requirements
1132
Check Resource Requirements for Software Object Against Set Top Box Resource Rights
1136
Authorized?
No
Yes
1140
Report Error to User
1144
Allow Object to Use Resource

Fig. 11

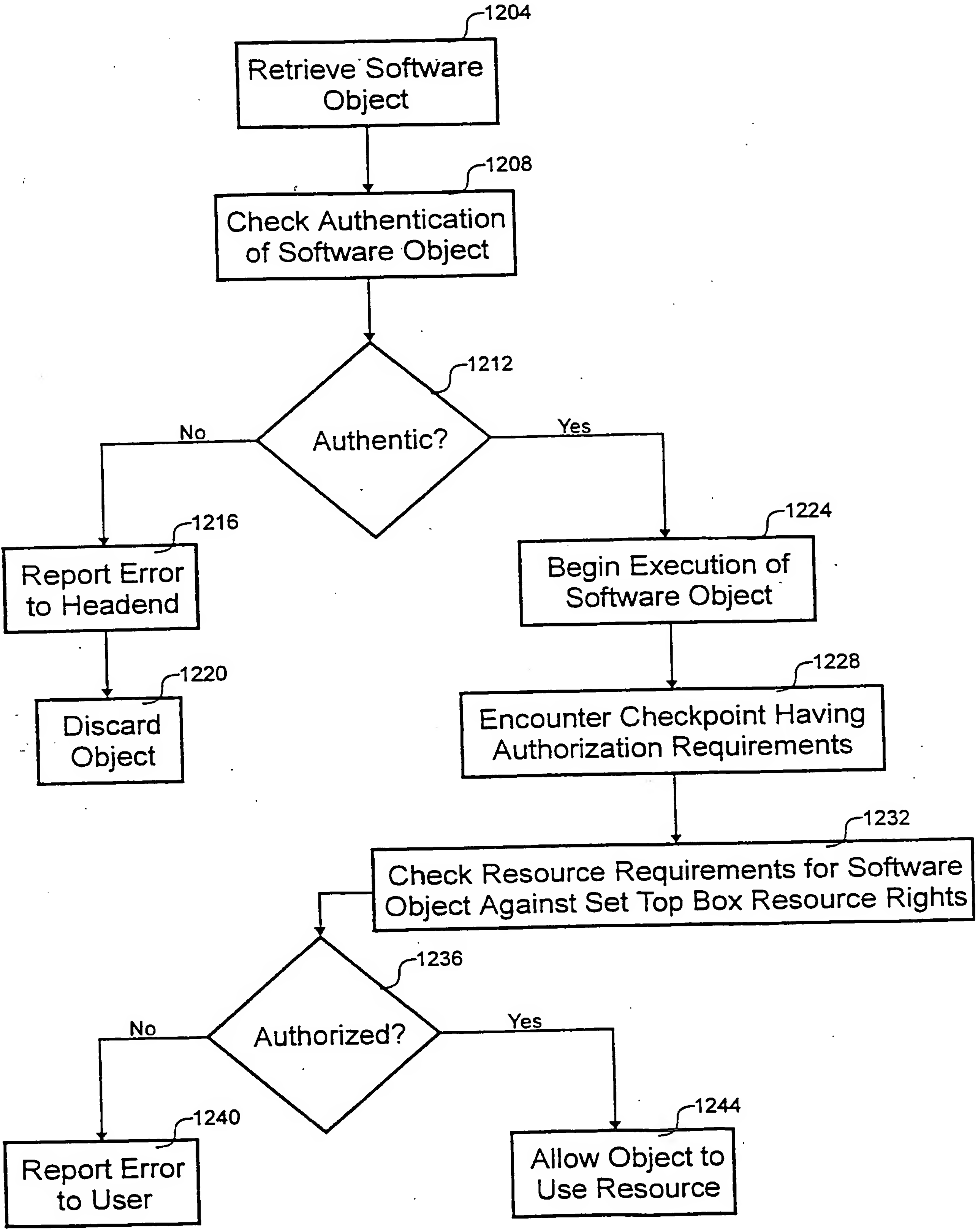

1204
Retrieve Software Object
1208
Check Authentication of Software Object
1212
Authentic?
No
Yes
1216
Report Error to Headend
1220
Discard Object
1224
Begin Execution of Software Object
1228
Encounter Checkpoint Having Authorization Requirements
1232
Check Resource Requirements for Software Object Against Set Top Box Resource Rights
1236
Authorized?
No
Yes
1240
Report Error to User
1244
Allow Object to Use Resource

Fig. 12

INTERNATIONAL SEARCH REPORT

Interi nal Application No
PCT/US 00/24097

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N5/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, PAJ, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 99 30217 A (GONG LI ;SUN MICROSYSTEMS INC (US)) 17 June 1999 (1999-06-17) abstract page 2, line 12 - line 36 page 3, line 6 -page 4, line 3 page 7, line 14 - line 22 page 9, line 18 - line 34 page 10, line 7 - line 22 page 14, line 4 - line 9	1-9, 11-16, 18-21
A	figures 2,5 ---	10,17

-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 21 December 2000	Date of mailing of the international search report 05/01/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Fantini, F

2

INTERNATIONAL SEARCH REPORT

Inter ›nal Application No
PCT/US 00/24097

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 0 909 094 A (CANAL PLUS SA) 14 April 1999 (1999-04-14) column 2, line 3 - line 15 column 3, line 32 - line 39 column 4, line 45 - line 56 column 5, line 17 - line 34 column 9, line 21 - line 33 ---	1-9, 11-16, 18-21
A	WALLACH D S ET AL: "Extensible security architectures for Java" PROCEEDINGS OF THE ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 1997, XP002101681 paragraphs '03.1!-'03.2! ---	1,8,10, 15,17,19
A	C. CRICHTON, J. DAVIES, J. WOODCOCK: "When to trust mobile objects: access control in the Jini Software System" TECHNOLOGY OF OBJECT-ORIENTED LANGUAGES AND SYSTEMS, 1999. TOOLS 30 PROCEEDINGS, 1 - 5 August 1999, pages 116-125, XP002155686 paragraphs '2.1.4!-'2.2.5! ---	1,8,15, 19
A	WO 98 55910 A (SUN MICROSYSTEMS INC) 10 December 1998 (1998-12-10) abstract page 12, line 3 - line 13 -----	1,8,15, 19

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/24097

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9930217 A	17-06-1999	US 6125447 A	26-09-2000
		AU 1718299 A	28-06-1999
EP 0909094 A	14-04-1999	AU 9363298 A	27-04-1999
		BR 9813024 A	15-08-2000
		EP 1021918 A	26-07-2000
		WO 9918730 A	15-04-1999
		NO 20001652 A	07-06-2000
		ZA 9809081 A	13-04-1999
WO 9855910 A	10-12-1998	US 5968136 A	19-10-1999
		EP 0934560 A	11-08-1999